君正 **Linux 2.4** 开发手册

Revision: 1.2
Date: Jan. 2008



君正 Linux 2.4 开发手册

Release history

Date	Revision	Change
Jan. 2008	1.2	Add description to section overview
Nov. 2007	1.1	Modify NAND flash filesystem description Add mplayer demo description Add guide to start the TFTP and NFS servers
May. 2007	1.0	First release

Disclaimer

This documentation is provided for use with Ingenic products. No license to Ingenic property rights is granted. Ingenic assumes no liability, provides no warranty either expressed or implied relating to the usage, or intellectual property right infringement except as provided for by Ingenic Terms and Conditions of Sale.

Ingenic products are not designed for and should not be used in any medical or life sustaining or supporting equipment.

All information in this document should be treated as preliminary. Ingenic may make changes to this document without notice. Anyone relying on this documentation should contact Ingenic for the current documentation and errata.

北京君正集成电路有限公司

北京市海淀区上地东路 **1** 号
盈创动力 **E** 座 **801C**
Tel: 86-10-58851002
Fax: 86-10-58851005
Http: //www.ingenic.cn

内容

1 概述

本文将向读者详细介绍基于君正处理器平台进行 Linux 2.4 内核开发的过程和方法，引导开发人员快速进行 Linux 开发，包括建立交叉编译环境、引导程序和 Linux 内核的配置和编译、设备驱动的测试和使用、根文件系统等等。

为了构建基于君正处理器的 Linux 2.4 的开发平台，您需要准备以下资源：

1） Linux 开发主机，用来安装编译器和相关源码
2） 基于君正处理器的开发板或产品板
3） MIPS 交叉编译工具链：君正提供基于 GCC-3.3.1 和 GLIBC-2.3.2 的工具链
4） 引导程序 U-Boot 源码：君正提供 u-boot-1.1.6 的源码
5） Linux 2.4 核心源码：君正提供 linux 2.4.20 核心的源码
6） 根文件系统：君正提供参考的根文件系统，基于 glibc 2.3.2 动态库
7） GUI 开发环境：用户需要根据自己产品构建 GUI 开发环境，如 QTE/GTK/MiniGUI 等

2 开发环境

2.1 安装交叉编译工具链

在君正处理器 JZ4730/JZ4740 平台上进行 Linux 2.4 内核开发之前，首先需要安装好 MIPS 的交叉编译环境。针对 Linux 2.4 内核开发，君正提供基于 GNU GCC 3.3.1 的 MIPS 交叉编译工具链，可分别安装在 Windows 和 Linux 主机上。Windows 上的交叉编译工具链需要在 Cygwin 下使用，因此，在 Windows 上进行开发前，请先安装好 Cygwin 环境。

在 Linux 主机上安装 MIPS 交叉编译工具的步骤如下：

首先，创建一工作目录，并把工具链包解压到该目录下，比如：

```
$ mkdir –p /opt
$ cd /opt
$ tar xzf mipseltools-gcc331-lnx24.tar.gz
```

其次，更新 PATH 环境变量：

```
$ export PATH=/opt/mipseltools-gcc331-lnx24/bin:$PATH
```

在 Windows 主机上安装 MIPS 交叉编译工具的步骤如下：

首先，安装和配置好 Cygwin 环境。请从网上下载 Cygwin 的相关软件包和资料，并参考 Cygwin 的文档进行安装。当安装好 Cygwin 后，启动 Cygwin 的 Shell 程序，并按照下面步骤进行安装：

```
$ mkdir –p /opt
$ cd /opt
$ tar xzf mipseltools-gcc331-lnx24-cygwin.tar.gz
$ export PATH=/opt/mipseltools-gcc331-lnx24/bin:$PATH
```

按照上面步骤建立好交叉编译环境后，可以编译简单的 helloworld.c 测试一下：

```
$ mipsel-linux-gcc –o helloworld helloworld.c
```

如果编译通过，说明刚刚安装的交叉编译工具链可以工作了。

强烈建议用户使用 Linux 做为开发主机。



2.2 启动 TFTP 和 NFS 服务

TFTP 是用来下载远程文件的网络传输协议，引导程序 U-Boot 支持 TFTP 下载功能。在开发阶段，如果主机启动了 TFTP 服务，引导程序 U-Boot 就可以通过 TFTP 下载 Linux 核心到目标板运行，这样将极大的方便用户进行开发。 因此，建议用户在用来开发的 Linux 或者 Windows 主机上启动 TFTP 服务。Linux 服务器上启动 TFTP 服务的步骤如下：

修改文件/etc/xinetd.d/tftp，主要是设置 TFTP 服务器的根目录和开启服务。修改后的文件如下：

```
service tftp
{
    disable = no
    socket_type  = dgram
    protocol     = udp
    wait         = yes
    user         = root
    server       = /usr/sbin/in.tftpd
    server_args  = -s /tftpboot
    per_source   = 11
    cps          = 100 2
    flags        = IPv4
}
```

然后创建/tftpboot 目录，启动 TFTP server，方法如下：

```
# service xinetd restart
```

这样，tftp-server 就启动了，你可以登陆本机进行测试。

网络文件系统（NFS）能够在不同的系统间实现文件的共享。 对于支持 NFS 客户端的操作系统如 Linux，在系统启动后可以通过 NFS 挂载根文件系统，这样可以运行位于 NFS 下的应用程序。在开发阶段，您需要配置一个 NFS 网络文件系统来为目标板提供根文件系统。当把编译好的应用程序安装在 NFS 的目录下后，就可以在操作系统启动后通过 NFS 挂载来运行该应用程序。Linux 服务器上启动 NFS 服务的步骤如下：

创建和编辑文件/etc/exports 为：

```
/nfsroot     *(rw,no_root_squash)
```

设定好后可以使用以下命令启动NFS:

```
# /etc/rc.d/init.d/nfs start
```

3 Linux 内核和驱动

君正移植了 CELinux（Linux 2.4.20）核心和所有的设备驱动程序，可以直接运行在君正处理器的各种开发平台上。本文将具体描述 CELinux 的内核和驱动程序。

君正 Linux 核心支持基于 Jz4730、 Jz4740 和 Jz4720 的各种开发平台。其中基于 Jz4730 的开发平台有：libra，gps，pmpv1 和 pmpv2 等；基于 Jz4740 的开发平台有 leo，pavo 等；基于 Jz4720 的开发平台有 virgo 等。

3.1 Linux 的目录结构

- arch/mips/：MIPS 体系相关目录和文件
  - kernel/：MIPS 内核相关文件
  - mm/：MIPS 内存管理相关文件
  - lib/：MIPS 公用库函数
  - jz4730/：JZ4730 处理器相关目录和文件
    - common/：JZ4730 处理器通用文件
    - libra/：LIBRA 开发板相关文件
    - pmpv1/：PMP ver 1.x 开发板相关文件
    - pmpv2/：PMP ver 2.x 开发板相关文件
    - gps/：GPS 开发板相关文件
  - jz4740/：JZ4740 处理器相关目录和文件
    - common： JZ4740 处理器通用文件
    - emurus：EMURUS FPGA 测试平台相关文件
    - leo： LEO 开发板相关文件
    - pavo：PAVO 开发板相关文件
    - virgo：Jz4720 VIRGO 开发板相关文件
  - ramdisk/：initrd 相关文件
  - uboot/：uImage 生成目录
  - zboot/：zImage 生成目录
  - config-shared.in：MIPS 共享配置文件
  - Makefile：MIPS 通用 makefile
  - defconfig-libra：LIBRA 开发板缺省配置
  - defconfig-pmpv1：PMP ver 1.x 开发板缺省配置
  - defconfig-pmpv2: PMP ver 2.x 开发板缺省配置
  - defconfig-gps: GPS 开发板缺省配置
  - defconfig-leo: LEO 开发板缺省配置
  - defconfig-pavo: PAVO 开发板缺省配置
- include/asm-mips/：MIPS 体系相关头文件
  - jz4730/：JZ4730 处理器相关头文件



  - jz4730-boards/：基于 JZ4730 目标板头文件
  - jz4740/：JZ4740 处理器相关头文件
  - jz4740-boards：基于 JZ4740 目标板头文件
- kernel：Linux 通用内核文件
- mm/：Linux 通用内存管理文件
- lib/：Linux 通用库函数
- init/：Linux 初始化函数
- ipc/：Linux 进程间通信函数
- net/：网络相关文件
- fs/：文件系统相关文件
  - jffs2/： JFFS/JFFS2 文件系统
  - yaffs2/: YAFFS/YAFFS2 文件系统
- drivers/：设备驱动目录
  - block/：块设备驱动
  - char/：字符设备驱动
  - char/jzchar/：JZSOC 字符设备驱动
  - dpm/：动态电源管理驱动
  - input/：输入设备驱动
  - mmc/：MMC/SD 卡驱动
  - mtd/：MTD 设备驱动
  - net/：网络设备驱动
  - sound/：音频设备驱动
  - ssfdc/：基于 NAND 的 USB Disk 驱动
  - ssi/：同步串行接口驱动
  - usb/：USB host 驱动
  - usb/gadget： USB device 驱动
  - video/：LCD framebuffer 驱动

3.2 配置和编译 Linux

首先，在一工作目录下解开 CELinux 的源码：

```
$ tar xjf celinux-040503.tar.bz2
```

然后，把 CELinux 的补丁打到源代码目录下：

```
$ cd celinux-040503
$ gzip -cd ../celinux-040503-jz-yyyymmdd.patch.gz | patch –p1
```

现在就可以配置和编译 Linux 了。

首先，选择目标板的配置，如下以 PMP version 2.x 平台为例：

```
$ make defconfig-pmpv2
```

然后，选择配置并保存：

```
$ make xconfig
```

最后，编译 Linux 核心：

```
$ make dep
$ make uImage
```

命令 'make uImage' 编译生成 U-Boot 可以引导的二进制映像 uImage，位于 celinux-040503/arch/mips/uboot/目录下。

3.3 Linux 配置选项

本节介绍君正 CELinux 的相关配置选项。

o 抢占式内核

君正 CELinux 支持抢占式内核功能，可以提高系统进程调度的实时性。配置 CELinux 选择 CONFIG_PREEMPT=y 来支持该功能，如下：

[General setup]

[•]y [ ]- [ ]n Preemptible Kernel

o 动态电源管理

君正 CELinux 支持动态电源管理(Dynamic Power Management)。DPM 提供了 CPU 变频和睡眠/唤醒的支持，以节省系统的功耗，选择以下配置：

[Power Management]

[•]y [ ]- [ ]n Power Management support

[•]y [ ]- [ ]n Enable Dynamic Power Management Support

[•]y [ ]- [ ]n Advanced Power Management (APM) Support

o **RTC 驱动**

配置选择 PCF8563 RTC 驱动，如下：

[Character devices]

[•]y [ ]- [ ]n Philips PCF8563 Real Time Clock (I2C Bus)



- o **MTD 驱动**

君正 CELinux 支持基于 NOR 和 NAND Flash 的 MTD 驱动，如下：

[Memory Technology Devices (MTD)]

[•]y [ ]- [ ]n Memory Technology Device (MTD) support
[•]y [ ]- [ ]n MTD partitioning support
[•]y [ ]- [ ]n Direct char device access to MTD devices
[•]y [ ]- [ ]n Caching block device access to MTD devices
[NAND Flash Device Drivers]
[•]y [ ]- [ ]n NAND Device Support
[•]y [ ]- [ ]n NAND Flash device on JZ4730 board
hardware_HM_ecc JZ4730 ECC Mode

- o **LCD Framebuffer 驱动**

君正处理器集成了 LCD 控制器，可以直接连接各种 TFT/STN 的 LCD 屏，配置 CELinux 如下：

[Console drivers]
[Frame-buffer support]
[•]y [ ]- [ ]n Support for frame buffer devices
[•]y [ ]- [ ]n JzSOC OnChip LCD controller support
[•]y [ ]- [ ]n SAMSUNG LTP400WQF01 TFT panel (480x272)

- o **AC97/I2S 音频驱动**

JZ4730 处理器集成了 AC97 和 I2S 音频控制器，如果选用 AC97，选择配置如下：

[Sound]
[•]y [ ]m [ ]n Sound card support
[•]y [ ]m [ ]n JzSOC AC97 sound

如果选用 I2S，选择配置如下：

[Sound]
[•]y [ ]m [ ]n Sound card support
[•]y [ ]m [ ]n JzSOC I2S sound
[AK4642EN] I2S codec type

- o **MMC/SD 卡驱动**

JZ4730 处理器集成了 MMC 和 SD 卡控制器，配置 CELinux 如下：

[MMC/SD Card support]

[•]y [ ]m [ ]n MMC support

[•]y [ ]m [ ]n JZ47xx SOC MMC/SD

o **USB Host 和 Device 驱动**

JZ4730 处理器支持 USB 1.1 的 HOST 和 DEVICE 控制器，CELinux 的配置如下：

[USB support]

[•]y [ ]m [ ]n Support for USB

[•]y [ ]m [ ]n OHCI(Compaq…) support

[•]y [ ]m [ ]n JzSOC OnChip OHCI-compatible host interface support

[Support for USB gadgets]

[ ]y [•]m [ ]n Support for USB Gadgets

[JZ4730-UDC] USB Peripheral Controller Driver

[ ]y [•]m [ ]n File-backed Storage gadget (DEVELOPMENT)

o **Camera 驱动**

JZ4730 处理器集成了 Camera 接口，支持 8 位数据线，支持 2048x2048 像数。CELinux 提供了通用的驱动程序接口，包含两个驱动：一个是 Camera 图像采集驱动，另一个是 sensor 的参数配置驱动，配置如下：

[Character devices]

[JzSOC char devices support]

[•]y [ ]m [ ]n JzSOC Camera Interface Module(CIM) support

[•]y [ ]m [ ]n JzSOC Common CMOS Camera Sensor driver

o **触摸屏驱动**

君正 CELinux 支持通用的触摸屏驱动，并支持不同的触摸屏接口类型，配置如下：

[Character devices]

[JzSOC char devices support]

[•]y [ ]m [ ]n JzSOC touchpanel driver support

[AK4182] Touchpanel codec type

o **SSI 串行总线驱动**

JZ4730 SSI 串行总线驱动配置如下：

[Synchronous Serial Interface]



[•]y [ ]m [ ]n Synchronous Serial Interface support

[•]y [ ]m [ ]n JzSOC SSI support

- **UART 串行口驱动**

[Character devices]

[•]y [ ]m [ ]n Virtual terminal

[•]y [ ]m [ ]n Support for console on virtual terminal

[•]y [ ]m [ ]n Standard/generic (8250/16550 and compatible UARTs) serial support

[•]y [ ]m [ ]n Support for console on serial port

- 以太网驱动

JZ4730 内置以太网驱动配置如下：

[Network device support]

[•]y [ ]m [ ]n Network device support

[Ethernet (10 or 100 Mbit)]

[•]y [ ]m [ ]n Ethernet (10 or 100 Mbit)

[•]y [ ]m [ ]n JzSOC On-Chip Ethernet Controller

- **JFFS2 文件系统**

君正 CELinux 支持 JFFS2 文件系统，配置如下：

[File systems]

[•]y [ ]m [ ]n Journaling Flash File System v2 (JFFS2) support

- **YAFFS2 文件系统**

君正 CELinux 支持 YAFFS2 文件系统，配置如下：

[File systems]

[YAFFS/YAFFS2 File systems]

[•]y [ ]m [ ]n YAFFS/YAFFS2 file system support

[•]y [ ]m [ ]n 512 byte / page devices

[•]y [ ]m [ ]n 2048 byte (or larger) / page devices

[•]y [ ]m [ ]n Autoselect yaffs2 format

[•]y [ ]m [ ]n Force chunk erase check

[•]y [ ]m [ ]n Cache short name in RAM

o 其它字符设备驱动

君正 CELinux 还提供了其它的设备驱动，如扫描键盘、UDC 热插拔和开关机管理等，配置如下：

[Character devices]

[JzSOC char devices support]

[•]y [ ]m [ ]n JzSOC scan keyboard support

[•]y [ ]m [ ]n JZ UDC hotplug driver support

[•]y [ ]m [ ]n JZ board poweroff support

4 Linux 根文件系统

Linux 内核编译好之后，还必须有根文件系统（root filesystem）才能使一个 Linux 系统正常运行。本节将简要介绍根文件系统的功能，以及如何根据自己系统的需求来制作根文件系统。

4.1 根文件系统的内容

根文件系统用于存放系统运行期间所需的应用程序、脚本、配置文件等，通常包含下面目录：

- bin/、sbin/：系统可执行程序和工具
- lib/：动态运行库
- etc/：系统配置信息和启动脚本 rcS
- usr/：用户可执行程序

4.2 制作根文件系统

嵌入式系统由于存储空间的限制，使得其对程序大小有严格的限制。特别是在制作根文件系统时，更要注意。而使用 BusyBox 就可以大大的简化根文件系统的制作。编译安装后的 BusyBox 只有一个二进制可执行文件 busybox，它实现了几乎所有常用、必须的应用程序（比如 init，shell，getty，ls，cp 等等），而这些应用程序都以符号链接的形式存在。对用户来说，执行命令的方法并没有改变，命令行调用会作为一个参数传给 busybox，即可完成相应的功能。使用 BusyBox 制作的根文件系统既可以节省大量的空间，还节省了大量的交叉编译的工作。

4.2.1 配置和编译 BusyBox

请登陆 BusyBox 的官方网站（http://www.busybox.net）下载起源码包。下面以 busybox-1.1.3 版本为例说明编译和安装 BusyBox 的过程。

编译 BusyBox 的过程类似于编译 Linux 内核的过程：

```
# make defconfig
# make menuconfig
# make
# make install
```

首先，对 BusyBox 进行配置。

make menuconfig



```
——————————————————————— BusyBox Configuration ———————————————————————
    Busybox Settings  ——->
——— Applets
    Archive Utilities  ——->
    Coreutils  ——->
    Console Utilities  ——->
    Debian Utilities  ——->
    Editors  ——->
    Finding Utilities  ——->
    Init Utilities  ——->
    Login/Passwd Management Utilities  ——->
    Linux EXT2 FS Progs  ——->
    Linux Module Utilities  ——->
    Linux System Utilities  ——->
    Miscellaneous Utilities  ——->
    Networking Utilities  ——->
    Process Utilities  ——->
    Shells  ——->
    System Logging Utilities  ——->
———
Load an Alternate Configuration File
Save Configuration to an Alternate File
```

```
Busybox Settings  ——->
    General Configuration  ——->
    Build Options  ——->
        [ ] Build BusyBox as a static binary (no shared libs)
        [*] Build with Large File Support (for accessing files > 2 GB)
        [*] Do you want to build BusyBox with a Cross Compiler?
        (mipsel-linux-) Cross Compiler prefix
        ( ) Any extra CFLAGS options for the compiler?
        [ ] Compile all sources at once
```

这里有几个选项要注意：动态编译，使用共享库；设置交叉编译器的路径和前缀，根据我们的情况就要设置为 mipsel-linux-。

```
    Install Options  ——->
        [*] Don't use /usr
        (./_install) BusyBox installation prefix
```

这里设置安装路径(./_install)。不使用/usr 的意思是指全部安装在/bin 和/sbin，不在/usr/bin 和/usr/sbin 下面安装程序。

Linux Module Utilities ——>

[*] insmod

[ ] In kernel memory optimization (uClinux only)

[*] rmmod

[*] lsmod

[*] modprobe

[*] Support version 2.2.x to 2.4.x Linux kernels

…

注意：配置 module 工具时不选择[In kernel memory optimization]，在编译模块.o 时不使用 GCC 的编译选项"-g"。

Shells ——>

Choose your default shell (ash) ——>

这里选择 ash 为缺省的 shell。

Networking Utilities ——>

[ ] Enable IPv6 support

[*] telnetd

[ ] Support call from inet only

这里的设置使得 BusyBox 可以支持 telnet server，这样可以通过远程 telnet 到目标机。

其余部分用来配置所实现的工具，可以根据需要选择，或者使用缺省配置即可。

然后就可以编译和安装了。



5 测试 Linux 内核和驱动

本节简单介绍如何测试君正 CELinux 集成的设备驱动程序，以验证驱动和硬件模块是否正常工作。

5.1 运行 Linux 内核

我们可以通过引导程序 U-Boot 启动 Linux 内核并挂载根文件系统来测试内核是否正常。首先配置和编译 Linux 内核，生成 uImage，通过 U-Boot 引导运行。如果 Linux 能正常启动并挂载成功根文件系统，说明 Linux 内核运行正常。

CELinux 常用命令行参数：

mem=xxM	设置内存容量大小
console=tty0	设置控制台输出为 tty0
console=ttyS0,57600n8	设置控制台输出为串口 ttyS0，波特率为 57600，8 个数据位，无校验位
ip=dhcp	IP 地址通过 DHCP 服务获取
ip=192.168.1.1	设置静态 IP 为 192.168.1.1
nfsroot=192.168.1.20:/nfsroot/pmp-root rw	设置网络根文件系统，具有读写权限
root=/dev/mtdblock0 rw	设置根文件系统为/dev/mtdblock0，具有读写权限
rootfstype=yaffs2	根文件系统类型为 YAFFS2

君正 CELinux 新增加的一些命令行参数有：

ts_debug	使能触摸屏测试模式

5.2 测试 Linux 设备驱动

o 测试 LCD 显示

配置 Linux 时选择好目标板 LCD 屏的型号，在启动 Linux 时就应该能看到 LCD 屏幕有图案输出。如果没有，请检查 LCD 屏的硬件连接。

o 测试音频播放和录音

MP3 播放测试：

```
# madplay test.mp3
```

录音和播放录音测试：

```
# vrec -S –s 48000 –b 16 sample1
# vplay –S –s 48000 –b 16 sample1
```

o　**测试视频播放**

测试经过优化的在 Jz4730 上运行的 mplayer:

```
# mplayer30 –vf scale=480:272 binhe.avi
```

测试经过优化的在 Jz4740 上运行的 mplayer:

```
# mplayer40 –vf scale=480:272 binhe.264
```

o　**测试 JFFS2 和 YAFFS2 文件系统**

配置 CELinux 时选择 MTD 设备，启动 Linux 后就可以测试基于 MTD 的文件系统 JFFS2 和 YAFFS2：

首先，查看 MTD 分区表信息：

```
# cat /proc/mtd
```

接着，格式化 MTD 分区：

```
# flash_eraseall –j /dev/mtd0     JFFS2 文件系统测试
# flash_eraseall /dev/mtd0     YAFFS2 文件系统测试
```

然后，挂载和测试文件系统：

```
# mount –t jffs2 /dev/mtdblock0 /mnt     JFFS2 文件系统测试
# mount –t yaffs2 /dev/mtdblock0 /mnt    YAFFS2 文件系统测试
# cp test /mnt
# umount /mnt
```

如果正常，说明文件系统已经工作。

o　**测试 MMC/SD 卡**

首先创建设备：

```
# mknod /dev/mmca b 243 0
# mknod /dev/mmca1 b 243 1
```



在 Linux 正常启动后，插入 MMC 或 SD 卡到 SD 卡座，Linux 就能自动探测到卡的插入并读出分区表信息，这时就可以挂载和操作 SD 卡了：

```
# mount –t vfat /dev/mmca1 /mnt/mmc
# ls /mnt/mmc
```

o　**测试触摸屏**

首先启动 Linux 前，Linux 命令行参数需要增加“ts_debug”选项。

启动 Linux 后，首先创建设备：

```
# mknod /dev/ts c 10 16
```

然后使用 cat 命令读该设备，当触摸屏有笔触时，就可以看到坐标值的输出：

```
# cat /dev/ts
```

o　**测试 USB Host**

首先，创建设备：

```
# mknod /dev/sda b 8 0
# mknod /dev/sda1 b 8 1
```

插入 USB 设备（如 U 盘）到 USB Host 端口，Linux 应能探测到设备的插入，这时可以挂载和操作 U 盘了：

```
# mount –t vfat /dev/sda1 /mnt/usb
# ls /mnt/usb
```

o　**测试 USB Device**

首先，以模块方式编译生成 USB Device 驱动：jz4730_udc.o 和 g_file_storage.o；然后按照下面方法加载驱动：

```
# insmod jz4730_udc.o
# insmod g_file_storage.o file=/dev/mtdblock5,/dev/mmca
```

这时我们挂载了两个 U 盘分区，一个是基于 NAND Flash 的分区 mtdblock5，另一个是基于 SD 卡的分区。现在用 USB 电缆连接 PC 和目标板的 USB Device 端口，PC 应该能识别到设备的插入并识别为 U 盘设备。

6 NAND Flash 文件系统

在消费类电子产品中，NAND Flash 得到越来越广泛地应用。君正处理器 JZ4730 和 JZ4740 都集成了 NAND Flash 控制器，支持与 NAND Flash 的直接连接。同时，JZ4730 处理器集成了硬件 HAMMING ECC 算法，可以检测出 256 字节数据中的二个比特错误位，并可纠正一个比特错误位。JZ4740 集成了硬件 REED-SOLOMEN ECC 算法，可以纠正 512 字节数据中至少 4 个比特位的错误。

君正 CELinux 也提供了对 NAND Flash 文件系统的支持，本节将介绍在 CELinux 上如何构建基于 NAND Flash 的文件系统。

6.1 NAND Flash 文件系统类型

CELinux 使用 MTD（Memory Technology Devices）驱动来管理 NAND Flash 设备。因为 JFFS2 和 YAFFS2 文件系统自己实现了 NFTL（NAND Flash Translation Layer）功能，所以在 MTD NAND Flash 设备上可以直接构建这两种文件系统。

君正还修改了 mtdblock 块设备驱动以支持 NAND Flash 的文件系统。修改后的 mtdblock 块设备驱动增加了逻辑块地址到物理块地址的映射、坏块管理、损耗均衡等功能。 这样，用户就可以在 NAND Flash 的 mtdblock 块设备上构建 EXT2、EXT3 和 FAT 等文件系统。

6.2 MTD 分区

在使用 NAND Flash 之前，首先要定义好 NAND Flash 的分区。MTD 分区的定义在 celinux-040503/drivers/mtd/nand/jz4730.c 或 jz4740.c 文件里。用户需要根据自己的系统分别定义 NAND Flash 物理块的分区信息。下面我们举 128MB 的 NAND Flash 为例来向读者介绍如何定义自己的分区：

NAND Flash 参数：

Page size：2048 Bytes
Block size: 128 KB
Pages per block: 64
Row address cycles: 3
Total size: 128MB
Total blocks: 1024

NAND 分区表信息：



分区	起始物理块	结束物理块	分区大小	分区描述
MTD BLOCK 0	0	23	3MB	MTD 分区 0 (bootloader and kernel)
MTD BLOCK 1	24	255	29MB	MTD 分区 1
MTD BLOCK 2	256	511	32MB	MTD 分区 2
MTD BLOCK 3	512	1023	64MB	MTD 分区 3

MTD 分区定义于文件 celinux-040503/drivers/mtd/nand/jz4730.c 或 jz4740.c 中，如下所示：

```
Static struct mtd_partition partition_info[ ] = {
    { name: “MTD BLOCK 0”,
      offset: 0 * 0x100000,
      size: 3 * 0x100000 },
    { name: “MTD BLOCK 1”,
      offset: 3 * 0x100000,
      size: 29 * 0x100000 },
    { name: “MTD BLOCK 2”,
      offset: 32 * 0x100000,
      size: 32 * 0x100000 },
    { name: “MTD BLOCK 3”,
      offset: 64 * 0x100000,
      size: 64 * 0x100000 }
}
```

6.3 创建 NAND Flash 文件系统

在 MTD 分区上创建 JFFS2 文件系统：

```
# flash_eraseall -j /dev/mtd1
# mount –t jffs2 /dev/mtdblock1 /mnt
# cp test /mnt
# umount /mnt
```

在 MTD 分区上创建 YAFFS2 文件系统：

```
# flash_eraseall /dev/mtd1
# mount –t yaffs2 /dev/mtdblock1 /mnt
# cp test /mnt
# umount /mnt
```

还可以用 mkyaffs2image 工具生成 YAFFS2 映像，然后用 nandwrite 命令写到 MTD 分区上，如下：

```
# mkyaffs2image 1 /rootfs/ rootfs.yaffs2
```

```
# nandwrite –a –o /dev/mtd1 rootfs.yaffs2
```

在 MTD 分区上创建 FAT 文件系统：

```
# mkfs.vfat /dev/mtdblock2
# mount –t vfat /dev/mtdblock2 /mnt
# cp test /mnt
# umount /mnt
```



7 Linux 电源管理

手持嵌入式设备由于电池寿命短，所以需要减少运行功耗和待机功耗。君正 CELinux 实现了电源管理功能，可以在系统运行时动态变频，并在系统长时间不使用时进入待机模式。其中，动态电源管理（DPM，Dynamic Power Management）可以允许系统在 50MHz 到 400MHz 范围内变频运行，而系统在待机模式下的电流可以降低到几毫安以下。下面将向读者介绍 CELinux 动态电源管理和待机模式的实现。

7.1 动态电源管理

CELinux 实现了君正处理器的动态电源管理，通过用户层制定策略与内核提供管理功能交互，可以实时调整 PLL 输出时钟、CPU 时钟、AHB 总线时钟、SDRAM 工作时钟和 APB 总线时钟等。

Linux 内核中与电源管理相关的代码位于代码树的 arch/mips/jz4730/common/ 和 arch/mips/jz4740/common/目录下，相关文件有 pm.c 和 dpm.c。

Linux 动态电源管理提供了操作系统级别的电源管理功能，但是需要用户层来制定和执行策略。DPM 后台进程 dpmd 就是用来监测系统的运行状况，并根据不同的状况设定 CPU 的工作频率的。

启动 dpmd 的步骤如下：

1） 配置 Linux 内核选择 CONFIG_DPM=y
2） 启动 DPM 初始化脚本/usr/local/bin/dpm_init
3） 根据系统修改 dpmd 的配置文件/usr/local/etc/dpmd.conf
4） 启动 dpmd

dpm_init 脚本用来初始化 DPM 系统，创建系统工作状态和策略。该脚本主要内容有：

1）初始化 DPM 系统

```
echo terminate > /proc/driver/dpm/cmd
echo init      > /proc/driver/dpm/cmd
```

2）创建 operating points

格式为：

```
create_opt <name> <Voltage> <NF> <NR> <NO> <ICLK_DIV> <SCLK_DIV> <MCLK_DIV>
<PCLK_DIV>
```

例如：

```
echo create_opt opt_58   0   190   2   1   6   12 12 12   > /proc/driver/dpm/cmd
```

3）创建 classes

例如：

```
echo create_class cls_58   opt_58   > /proc/driver/dpm/cmd
```

4）创建 policies

例如：

```
echo "create_policy 58 cls_58 cls_58 cls_58 cls_58" \
                         "cls_58 cls_58 cls_58 cls_58" \
          "cls_58" \
          "cls_58 cls_58 cls_58 cls_58" > /proc/driver/dpm/cmd
```

配置文件/usr/local/etc/dpmd.conf 定义了系统运行状态的参数，dpmd 将根据这些参数来设置 CPU 的工作频率。现分别介绍如下：

POLICY_HIGH=XXX

DPM 初始化脚本定义了多个 POLICY，'XXX'必须是其中的一项。POLICY_HIGH 定义了 CPU 最高工作频率对应的 POLICY。

POLICY_LOW=XXX

DPM 初始化脚本定义了多个 POLICY，'XXX'必须是其中的一项。POLICY_LOW 定义了 CPU 最低工作频率对应的 POLICY。

POLL_INTERVAL=N

POLL_INTERVAL 定义了 dpmd 读取和计算系统运行状态的时间间隔，单位是"秒"。如设置"POLL_INTERVAL=2"表示每 2 秒钟计算一次系统运行状态，并根据状态值设定合适的 CPU 工作频率。

CPU_INTERVAL=M-N
SUSPEND_INTERVAL=N
PROGRAMS=XXX,XXX,XXX

CPU_INTERVAL/SUSPEND_INTERVAL/PROGRAMS 分别定义不同的规则，dpmd 将根据这些规则



来计算系统的运行状态。

CPU_INTERVAL 定义了 CPU 工作于 POLICY_LOW（最低工作频率）的 CPU 负载值，'M'-'N'为 CPU 负载的百分比值。如定义“CPU_INTERVAL=0-15”表示当 CPU 负载位于 0%和 15%之间时，CPU 将工作于最低工作频率。当 CPU 负载大于 CPU_INTERVAL 定义的数值时，CPU 将工作于 POLICY_HIGH(最高工作频率)。

SUSPEND_INTERVAL 定义了 CPU 进入睡眠所需的 IDLE 时间间隔，单位为“秒”。当 CPU 负载位于 CPU_INTERVAL 所定义的数值之间时表示 CPU 处于 IDLE 状态。当 IDLE 时间间隔大于 SUSPEND_INTERVAL 定义的数值时，CPU 将进入睡眠状态。

PROGRAMS 定义了 CPU 工作于 POLICY_HIGH（最高工作频率）的进程。可以定义多个进程，不同进程用逗号“,”隔开。如定义“PROGRAMS=mpegplayer,madplay”表示当系统运行进程 mpegplay 或 madplay 时，CPU 将工作于最高工作频率。

dpmd 使您可以根据自己系统的需要来定义不同的规则。

7.2 系统睡眠和唤醒

系统如果长时间处于闲置状态时，可以让其进入睡眠模式。在这种模式下，系统大部分模块都置于低功耗模式，SDRAM 处于自刷新模式并保存程序运行的现场，只保留 RTC 时钟工作以唤醒系统。JZ4730 处理器睡眠和唤醒的流程如下：

1） 系统根据运行状态进入睡眠模式；
2） 操作系统根据 PM 机制调用各个设备驱动的 suspend 函数让设备进入睡眠模式；
3） 操作系统保存 CPU 现场状态；
4） CPU 执行睡眠指令进入睡眠模式，并让 SDRAM 进入自刷新模式；
5） 通过按键或者其它中断唤醒 CPU；
6） CPU 被唤醒并 reset；
7） 根据 reset 状态寄存器判断当前为 hibernate 唤醒复位，CPU 跳到 resume 函数恢复睡眠前的系统状态，并调用各个设备驱动的 resume 函数让设备恢复工作；
8） 系统正常运行；

用户程序通过执行下面命令进入睡眠模式：

```
echo 1 > /proc/sys/pm/suspend
```